ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcs228932020.pdf
https://doi.org/10.30534/ijatcse/2020/228932020

Reconfigurable Design of Low Power Hybrid Crypto Processor using Signcryption for Wireless Networks

C.Lakshmi[1], P.Jesu Jayarin[2]
[1]Reasearch Scholar, Sathyabama Institute of Science and Technology,India, c.lakshmichandrasekar@gmail.com
[2]Associate Professor,Department of Information Technology,Jeppiaar Engineering College,India,
,jjayarin@gmail.com

ABSTRACT

Now a day all the communications are carried out in wireless medium. It is necessary to transmit the confidential data in wireless media in secure manner. Cryptography is technique to protect electronic data in communication network. Hardware implementation of cryptography processor in field programmable gate array (FPGA) is major issues in terms of area, power and throughput. In this paper, we propose a hybrid crypto processor (HCP) for wireless network using flexible encryption and signature techniques. The main aim of proposed HCP technique used to provides secure wireless communication with aware of malicious attacks in the network, know to my knowledge, the proposed HCP hardware design is open the platform in signcryption. Generally, hybrid cryptography consists of data encapsulation mechanism (DEM) and key encapsulation mechanism (KEM). In HCP design, the efficient multiplier based ECC processor is proposed for data encapsulation over Galois field (GF (2^m)). Moreover, the improved enhanced Kurosawa and Desmedt hashing (EKD) hashing scheme is proposed for key encapsulation. The proposed HCP design is implementing in Xilinx tool with different field programmable gate array (FPGA) families. The result shows that effectiveness of proposed HCP design in terms of hardware utilization, power consumption and throughput over existing design.

Key words : Cryptography, ECC processor , EKD hashing , hybrid crypto processor

1. INTRODUCTION

Confidentiality, integrity, non-repudiation and authentication are important issues in information security. In order to achieve these security services in many applications both encryption and digital signature are required [1]. In public key encryption, a message is digitally signed and then followed by an encryption also denotes as “signature-then encryption” [2]. In signature-then-encryption there are two problems are high cost and low efficiency. In order to solve the problems that exist in signature-then-encryption, the signcryption was offered [3]. Signcryption is very new technique which is supposed to fulfill the functionalities of digital signature and encryption Signcryption is a new concept in public key cryptology. It provides a common framework for a number of protocols which are used to provide a confidential and authenticated transmission channel for messages. One of the best properties of signcryption is capability of providing both encryption and digital signature at the same time. In other words, by using signcryption an acquire confidentiality, authentication, integrity, unforgeability, non-repudiation, public verifiability and forward secrecy of message confidentiality [4].

Generally, hybrid signcryption is easy implement in software, but typically too slow for real-time applications, such as storage devices, embedded systems, network routers, etc. For this reason, it becomes necessary to implement on hardware [8]. The hardware cryptographic systems must fulfill contradictory requirements are fast parallel structures implementing computationally extensive cryptographic functions must coexist with complex sequential structures used to implement cryptographic algorithms such as cipher modes, key management operations and cryptographic protocols [9]. Implementation of cryptographic algorithms and protocols in hardware necessitates employing many complex state machines that make the logic vulnerable. Furthermore, upgrades of hardwired logic can become complicated, long and expensive but security of the system itself and protection of confidential data is often underestimated. Recently, cryptographic algorithms were more frequently implemented in FPGAs using SRAM and flash-based devices [10]. SRAM-based devices keep their configuration in volatile configuration SRAM, so the device has to be configured after every power-up. In contrast, Flash-based FPGAs store their configuration in internal flash memory so device configuration does not have to be configured after power-up again. FPGAs are very suitable for many cryptographic algorithms. Because of their high parallelism, high-performance data security algorithms can be significantly accelerated when compared to software implementations [11]. FPGAs can be reprogrammed therefore hardware updates are cheap and easy to perform in place or even remotely.

The hardware implementation of FPGA with SHA-512 [12] is too complex and consumes very high hardware cost. Here, an error detection scheme solves the problem based on information as well as hardware redundancy schemes. Errors in most of the operations in a digest round are detected by simple parity circuits and errors. The modified Itoh-Tsujii inverse algorithm (ITA) [13] algorithm implemented on FPGA with the better and requires shorter addition chain of resources. FPGA-based SHA-1 cryptanalysis system [14] used to achieve an EOC much higher than other existing software and hardware solutions. The keyed-hash message authentication code (HMAC) is scheme [15] used to achieve fault tolerance in the secure hash standard (SHS) and also reduces implementation area requirements and power consumption. The hardware implementation of the RC4 algorithm [16] based on dual-port block RAM in the FPGA in order to better utilize the available logic and memory resources, which achieves better performance. A high throughput digital design of the 128-bit advanced encryption standard (AES) algorithm [17] is based on the C-slow retiming, which provides design with feedback loops and automatically rebalances the registers in the design. Area-efficient, high-throughput multi-mode architectures for the SHA-1 and SHA-2 hash families are designed by a systematic flow for designing multi-mode architectures [18].

Recently, elliptic curve cryptography (ECC) plays the importance role in security solution. ECC offers guaranteed security with smaller key size, faster cryptographic operations and running on smaller chips. Efficient, compact hardware implementations are available for ECC process with the smaller key length. Memory architecture for elliptic curve cryptography (ECC) [19] with multiple modular multiplier units are suited for different point addition and doubling algorithms over prime field implemented on FPGAs. ECC allows the execution time to scale with the number of modular multipliers and exhibits nearly no overhead compared to the mere runtime of the multipliers.

J, Sasi Bhanu et al [30] proposed the IoT enhancing the performance of IoT model by using high performance computing. Since IoT uses various hardware equipment like sensors,processors. The processing speed need to be improved in certain application.

Now a days the data is transmitted in remote location by using cloud. The security of information is a sensitive factor in cloud database[33].In this paper [33] the Distributed Denial of Service attack is detected in cloud and security mechanism is provided

2. RELATED WORK

Hardware design [21], [23],[24],[25],[26],[27] authors can maintain the high throughput while offering the robustness and reliability but usually involve the highest cost. The application specific and modular hardware design [28],[29],[31] can achieves extraordinary power and offers high reliability but comes with higher cost due to the need of additional hardware and lost speed of operation. The recent high speed ECC processor [32] uses the single and three multipliers to optimize the throughput of design, but the hardware utilization of this processor is very high due to the three multipliers. Three multipliers reduce latency to speed up the point operation; however, the critical path delay is increased.

The problems can overcome by a hybrid crypto processor (HCP) based on flexible configuration of encryption and signature techniques. To the best of our knowledge our HCP design is the first hardware architecture for signcryption. Figure 1 shows the basic system model of proposed hybrid crypto processor. It utilizes EKD hash scheme used for the KEM; and ECC processor used for DEM. Typically HCP design provides the cipher text should reveal no meaningful information about the original message, even to an active adversary who can probe a decryption oracle with chosen cipher texts. The ECC processor consists of three units such as memory unit, control unit and arithmetic unit. In this, we concentrate to optimize the arithmetic unit by the efficient multiplier instead of single and three multipliers in [32]. The flexible design supports all field size of recommended GFs without the need to reconfigure the hardware, which is very opt for hardware efficient design. The main contribution of proposed EHSP design is summarized as follows:

1. In HCP design, EKD hash scheme used to encapsulate the key and ECC processor used to encapsulate the data. ECC processor consists three units are memory, control and arithmetic unit; here we enhance the arithmetic unit because it consumes more hardware cost and time.
2. Proposed HCP design synthesized and implemented on the Virtex4, Virtex5 and Virtex7 FPGA family; and compare it performance with the existing crypto processor in terms of hardware utilization, energy consumption and throughput.

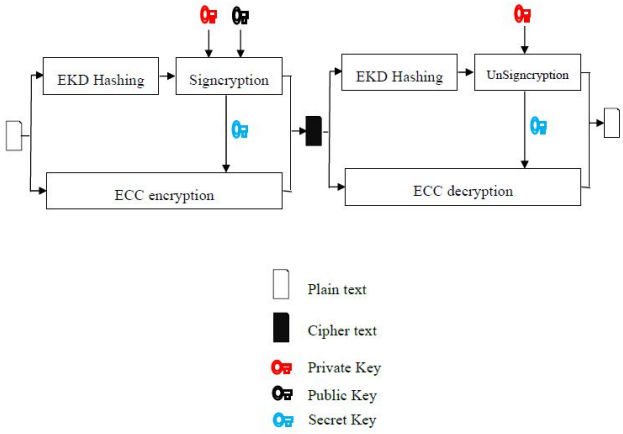

Figure 1: System model of proposed hybrid cryptography processor (HCP)

3.HYBRID CRYPTO PROCESSOR

The working function and the mathematical model of HCP design with an ECC processor and MKDH scheme is briefly describe as follows.

3.1 Data encapsulation using ECC processor

A data encapsulation mechanism (DEM) employs the symmetric key from signcryption KEM to encrypt the message of arbitrary length. ECC is able to provide the same cryptographic strength as Rivest–Shamir–Adleman (RSA)-based system with much smaller key sizes. For example, a 256 bit ECC key is equivalent to RSA 3072 bit keys. The latest, most secure symmetric algorithms use at least 128 bit keys, so it makes sense that the asymmetric keys provide at least this level of security.

The ECC processor is shown in Figure 2, it consists of major three units are memory, control and arithmetic unit. Here, we concentrate on arithmetic unit, because, it consumes more hardware than other units.

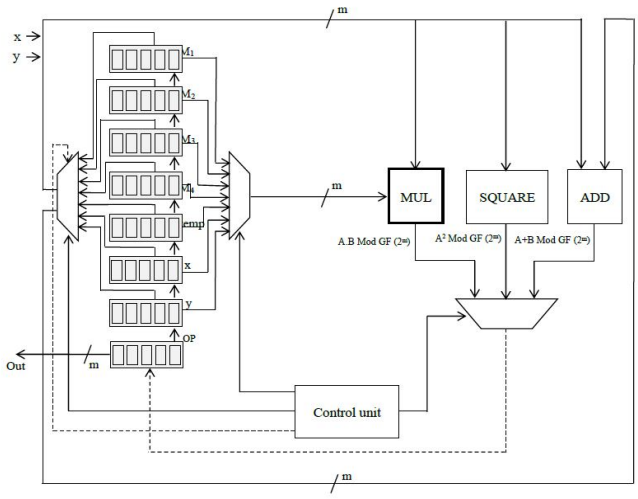

Figure 2:Hardware Architecture of ECC Processor

The proposed multiplier is shown in figure 3 uses two high speed adders such as MBM and Wallace tree multiplier which are hybridized with CLA to perform the final accumulation of the partial products[35]. The multiplication process consists of three steps. They are: 1) generate the partial products; 2) add the generated partial products until the last two rows are remained; 3) compute the final multiplication results by adding the last two rows. The modified Booth algorithm reduces the number of generated partial products by half in the first step.

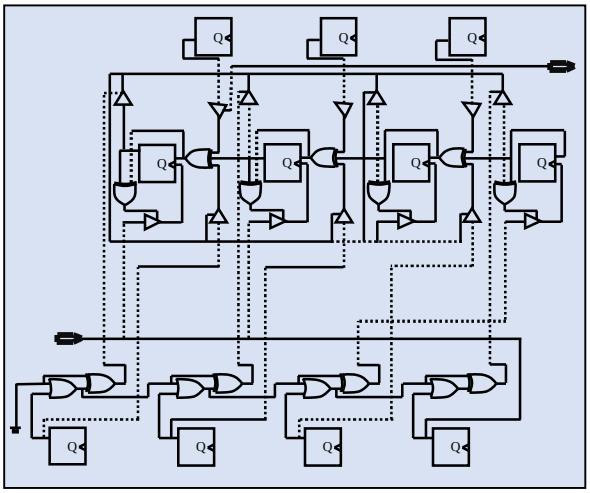

Figure 3: Flexible Multiplier

3.2 Modified Kurosawa and Desmedt hashing (MKDH) scheme

Key encapsulation mechanism (KEM) [34] is an asymmetric encryption structure licenses setting aside a couple of minutes a without question key together with encryption. The exemplified key will be used for long data symmetric encryption, while the exemplification is used for sharing keys. In standard model, the relationship of a symmetric key K_s contains gathering ranges. Here, we split into two free keys and K_s^2. The key K_s^2 is inside used to see the epitome and the key K_s^1 is then returned as the focal key. It wires key age, structure, and decapsulation. The key age process handles the security parameter k, open key (K_{pb}), public key (K_s) and encryption (E) used to returns $(CT,\ K)$. Consider an element s_1 randomly from a set S is rotationally expressed by $s_1 \overset{\$}{\leftarrow} S$. Let k be the security parameter, the target collision resistant hash function denotes as $TCR : \varepsilon(k) \to \mathrm{O}(k)$ with a target of $s_1^* \overset{\$}{\leftarrow} \varepsilon(k)$, the situation is hard for all poly-time set S to find $s_1 \in \varepsilon(k)$ and satisfying $TCR(s_1) = TCR(s_1^*)$.

$$\mathfrak{N}_S^{TCR}(k) = Pr[s_1 \leftarrow S(s_1^*) : s_1 \neq s_1^* \wedge TCR(s_1) = TCR(s_1^*)] \quad (1)$$

Keyderivations function $KDF : \mathrm{K}(k) \to \{0,1\}^{2n(k)}$ such that random $K \in \mathrm{K}(k)$ is computationally random over $\{0,1\}^{2n(k)}$ by

$$\Re_d^{KDF}(k) = \left| \Pr_{K_E \leftarrow K(k)} \left[d\left(KDF(K_E)\right) = 1 \right] - \Pr_{(k,k') \leftarrow \{0,1\}^{2n(k)}} \left[d(k,k') - 1 \right] \right| \quad (2)$$

where d represents the poly-time distinguishers.

A message authentication code $MAC : \{0,1\}^{n(k)} \times \varepsilon(k) \rightarrow \{0,1\}^{m(k)}$ used to compute the target by $T = MAC_k(s_1)$ and the details as follows:

$$\Re_S^{MAC}(k) = \Pr\left[s_1 \neq s_1^* \wedge T = MAC_c(s_1) \right] \quad (3)$$

Let $G = \langle g \rangle$ be a get-together, made by g, of prime open enthusiasm $2^k < q < 2^{k+1}$ for security parameter k. The Diffie Hellman question on G declares that, for all poly-time distinguishers d, non-unit subjective areas $g_1, g_2 \overset{\$}{\leftarrow} G$ and $r \neq s \overset{\$}{\leftarrow} Z_q$. The key age takes after $(s_1, s_2, s_3, s_4) \overset{\$}{\leftarrow} Z_q^4$ with the control qualities $x = g_1^{s_1}, g_2^{s_2}$ and $y = g_1^{s_3}, g_2^{s_4}$, which impacts open, to problem keys as takes after:

$$K_{pb} \leftarrow (g_1, g_2, x, y) \quad (4)$$

$$K_E \leftarrow (s_1, s_2, s_3, s_4) \quad (5)$$

In key encapsulation, K_{pb} used to define the secret b

$$K_E \leftarrow C^r d^{r\alpha} \quad (6)$$

where the parameter α represents the $TCR(g_1^s, g_2^s)$. The probability of advance in KD hash is to spilt the puzzler keys into two individual parts, MAC process gives $T = MAC_{K_s^2}(g_1^r, g_2^r)$, finally restores the figure substance and key as

$$CT = (g_1^r, g_2^r, T) \text{ and } K = K_E^1 \quad (7)$$

In decapsulation, if $g_1^r \notin G$ or then $g_2^r \notin G$ $\perp$ is returned quickly toward the beginning stage. The exemplified key (K) will forward to the encryption figuring and amassed check point.

The hardware architecture of MKDH function is shown in the figure 4 simple iterated construction with a variable-length input and arbitrary length output based on a fixed length transformation operating on a fixed number of bits. The fixed number of bits is the width of the permutation, or bit state. The bit state is the sum of bit rate (m) and bit capacity (c). Before any permutation is performed, MKDH initializes state s and pads the message to make the string a multiple of m. The padded message, represented by P, is then divided into i blocks. For sponge construction, there are two phases.

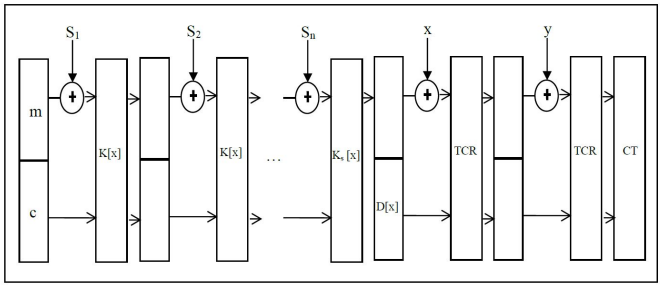

Figure 4: Hardware architecture of MKDH function

All the rounds of MKDH perform identical operations; except for using different CT. MKDH is designed such that the dependence on CPU word length is only due to fixed rotations, leading to an efficient use of CPU resources on a wide range of processors. The default bit state for MKDH permutation (1,600) is chosen in favor of 163-bit architectures

4. SIMULATION RESULTS

The proposed efficient hybrid cryptography processor (HCP) has been implemented in Verilog HDL. For simulation, synthesis, mapping, and routing purposes Xilinx ISE 14.5 design suite has been used. The design was implemented on Virtex4 (XC4VLX60), Virtex5 (XC5VLX50), and Virtex7 (XC7V330T) families to allow for a fair comparison.The RTL schematic of HCP design is showed in figure 5.

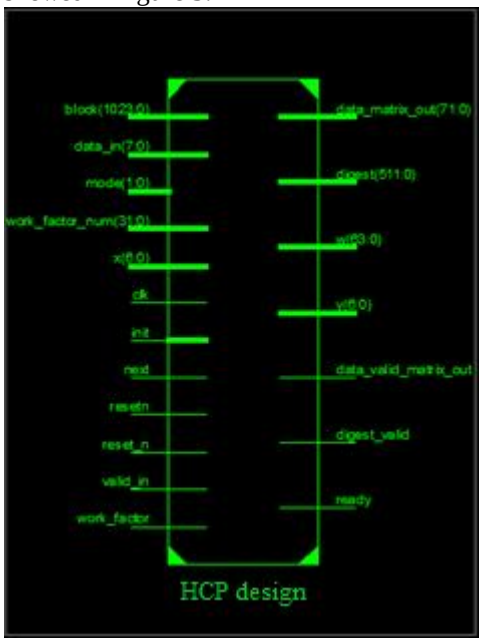

Figure 5:RTL Schematic of Hybrid crypto Processor

4.1.Hardware Utilizaion Comparision

After design implementation, we can verify the device utilization by reviewing the design summary. Here, proposed EHSP design is compared with the existing cryptography processors such as ECC processor using radix-4 booth encoding based interleaved modular multiplier [21], ECC processor using double scalar multiplier [22], dual field ECC processor [23], 128 bit AES [25], hybrid AES + Grøstl hash [27], RNS based ECC [28] and low latency ECC processor [31]. Table 1 shows the performance comparison in terms of resource utilization for proposed and existing schemes. Table describes the number of slices, FFs and LUTs comparison of proposed EHSP design and low latency ECC processor [32] over GF (2^{163}) in terms of pictorial representation. For m=163, the proposed EHSP (hybrid) design consumes hardware equivalent or little bit higher than single ECC design [32]. Compare to other designs the hybrid architecture consumes same amount of hardware, but provides double security. The comparision is showed in Table 1.

Table 1: Performance comparison with exiting FPGA implementations

DESIGN	FIELD SIZE (BITS)	PLATFORM	DEVICE UTILIZATION		
			SLICES	FLIP-FLOPS	LOOKUP TABLES (LUTS)
[21]	256	Virtex4	13158	-	-
		Virtex6	11104	-	-
[22]	256	Virtex7	377	992	995
[23]	256	Virtex2	-	-	12425
[25]	128	Virtex5	2940	-	-
		Virtex6	2537	-	-
		Virtex7	2617	-	-
[27]	128&256	Virtex6	102	-	
[28]	256	Virtex6	1620	-	3360
		Virtex7	1558	-	3370
[32]	163	Virtex4	12964	3077	23468
		Virtex5	4393	3090	16090
		Virtex7	4150	3747	14202
	571	Virtex7	50336	29217	141078
EHSP*	163	Virtex4	2332	916	4105
		Virtex5	982	543	3629
		Virtex7	916	540	1465

***Proposed Design**

4.2 Maximum Frequency Comparison

Maximum clock frequency of proposed EHSP design is compared with the existing cryptography processors are ECC processor using radix-4 booth encoding based interleaved modular multiplier [21], ECC processor using double scalar multiplier [22], dual field ECC processor [23], 128 bit AES [25], hybrid AES + Grøstl hash [27], and low latency ECC processor [31]. Table 2 shows the performance comparison in terms of maximum frequency achieved from the proposed and existing schemes. It describes maximum frequency comparison of proposed EHSP design and low latency ECC processor [31] over GF (2^{163}) with different FPGA families. For m=163, the proposed EHSP (hybrid) design achieves maximum clock operating frequency than existing processor [32].

Table 2: Maximum Frequency Comparison With Exiting Fpga Implementations

DESIGN	FIELD SIZE (BITS)	PLATFORM	MAXIMUM FREQUENCY (MHZ)
[21]	256	Virtex4	40
		Virtex6	70
[22]	256	Virtex7	205.634
[23]	256	Virtex2	55.7
[25]	128	Virtex5	704.7
		Virtex6	740.7
		Virtex7	81.3
[27]	128&256	Virtex6	413
[32]	163	Virtex4	210
		Virtex5	228
		Virtex7	352
	571	Virtex7	111
HCP*	163	Virtex4	191.119
		Virtex5	224.941
		Virtex7	373.756

4.3 Power Comparison

The power consumption of proposed EHSP design is compared with the existing cryptography processors are ECC processor using radix-4 booth encoding based interleaved modular multiplier [21] and ECC processor using double scalar multiplier [22]. Table 3 shows performance comparison in terms of power consumption achieved from proposed and existing schemes.For m=256 bits, proposed EHSP (hybrid) design consumes very less power compare to the simple processors.

Table 3: Power Consumption Comparison

DESIGN	FIELD SIZE (BITS)	PLATFORM	POWER CONSUMPTION (MW)
[21]	256	Virtex4 Virtex6	173 192
[22]	256	Virtex7	208
HCP*	163	Virtex4 Virtex5 Virtex7	102 109 143

5 .CONCLUSION

In this paper, we have proposed an efficient hybrid cryptography processor (HCP) using the modified key encapsulation method (KEM) and data encapsulation method (DEM). The modified Kurosawa and Desmedt hashing (MKDH) scheme used for KEM and the elliptic curve cryptography (ECC) processor used for DEM. The flexible data encryption is achieved by the flexible multiplier over GF (2^{751}) without reconfigurable manner. Results proves effectiveness of proposed HCP design in terms of less hardware utilization, power consumption and high maximum frequency in different FPGAs Virtex4 (XC4VLX60), Virtex5 (XC5VLX50) and Virtex7 (XC7V330T). Moreover, the proposed EHSP design provides high security by maximize the lifetime of key. In future, this design can suitable to applied for wireless networks for secure data transmission.

REFERENCES

[1]H. Petersen and M. Michels, "**Cryptanalysis and improvement of signcryption schemes**", IEE Proceedings - Computers and Digital Techniques, vol. 145, no. 2, p. 149, 1998.
https://doi.org/10.1049/ip-cdt:19981862

[2]W. He and T. Wu, "**Cryptanalysis and improvement of Petersen–Michels signcryption scheme**", IEE Proceedings - Computers and Digital Techniques, vol. 146, no. 2, p. 123, 1999.

[3]Q. Huang, D. Wong and G. Yang, "**Heterogeneous Signcryption with Key Privacy**", The Computer Journal, vol. 54, no. 4, pp. 525-536, 2011.

[4]F. Li, H. Zhang and T. Takagi, "**Efficient Signcryption for Heterogeneous Systems**", IEEE Systems Journal, vol. 7, no. 3, pp. 420-429, 2013.
https://doi.org/10.1109/JSYST.2012.2221897

[5]C. Hu, N. Zhang, H. Li, X. Cheng and X. Liao, "**Body Area Network Security: A Fuzzy Attribute-Based Signcryption Scheme**", IEEE Journal on Selected Areas in Communications, vol. 31, no. 9, pp. 37-46, 2013.

[6]W. LIU and C. XU, "**Certificateless Signcryption Scheme Without Bilinear Pairing**", Journal of Software, vol. 22, no. 8, pp. 1918-1926, 2011.

[7]Y. Zhang, X. Chen and H. Li, "**Key-Evolving Hierarchical ID-Based Signcryption**", The Computer Journal, vol. 56, no. 10, pp. 1228-1248, 2012.

[8]Yi Wang, J. Leiwo, T. Srikanthan and Yu Yu, "**FPGA based DPA-resistant Unified Architecture for Signcryption**", Third International Conference on Information Technology: New Generations (ITNG'06), 2006. https://doi.org/10.1109/ITNG.2006.66

[9]Peng Changgen and Li Xiang, "**Threshold signcryption scheme based on elliptic curve cryptosystem and verifiable secret sharing**", Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005..

[10]A. Elbirt, W. Yip, B. Chetwynd and C. Paar, "**An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists**", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 9, no. 4, pp. 545-557, 2001.

[11]Y. Xiaohui, D. Zibin, L. Yuanfeng and W. Ting, "**Researching and implementation of reconfigurable Hash chip based on FPGA**", Journal of Systems Engineering and Electronics, vol. 18, no. 1, pp. 183-187, 2007.

[12]I. Ahmad and A. Das, "**Analysis and Detection Of Errors In Implementation Of SHA-512 Algorithms On FPGAs**", The Computer Journal, vol. 50, no. 6, pp. 728-738, 2007.
https://doi.org/10.1093/comjnl/bxm023

[13]C. Rebeiro, S. Roy, D. Reddy and D. Mukhopadhyay, "**Revisiting the Itoh-Tsujii Inversion Algorithm for FPGA Platforms**", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 8, pp. 1508-1512, 2011.

[14]A. Cilardo and N. Mazzocca, "**Exploiting Vulnerabilities in Cryptographic Hash Functions Based on Reconfigurable Hardware**", IEEE Transactions on Information Forensics and Security, vol. 8, no. 5, pp. 810-820, 2013.

[15]M. Juliato and C. Gebotys, "**A Quantitative Analysis of a Novel SEU-Resistant SHA-2 and HMAC Architecture for Space Missions Security**", IEEE Transactions on Aerospace and Electronic Systems, vol. 49, no. 3, pp. 1536-1554, 2013.

[16]E. Taqieddin, O. Abu-Rjei, K. Mhaidat and R. Bani-Hani, "**Efficient FPGA Implementation of the RC4 Stream Cipher using Block RAM and Pipelining**", Procedia Computer Science, vol. 63, pp. 8-15, 2015.
https://doi.org/10.1016/j.procs.2015.08.306

[17]R. Farashahi, B. Rashidi and S. Sayedi, "**FPGA based fast and high-throughput 2-slow retiming 128-bit AES encryption algorithm**", Microelectronics Journal, vol. 45, no. 8, pp. 1014-1025, 2014.

[18]H. Michail, G. Athanasiou, G. Theodoridis and C. Goutis, "**On the development of high-throughput and area-efficient multi-mode cryptographic hash designs in**

FPGAs", Integration, the VLSI Journal, vol. 47, no. 4, pp. 387-407, 2014.
https://doi.org/10.1016/j.vlsi.2014.02.004
[19]R. Laue and S. Huss, "**Parallel Memory Architecture for Elliptic Curve Cryptography over GF (P) Aimed at Efficient FPGA Implementation**", Journal of Signal Processing Systems, vol. 51, no. 1, pp. 39-55, 2007.
[20]Y. Dan, X. Zou, Z. Liu, Y. Han and L. Yi, "**High-performance hardware architecture of elliptic curve cryptography processor over GF(2163)**", Journal of Zhejiang University-SCIENCE A, vol. 10, no. 2, pp. 301-310, 2009.
[21]K. Javeed and X. Wang, "**FPGA Based High Speed SPA Resistant Elliptic Curve Scalar Multiplier Architecture**", 2018. .
[22]Z. Liu, J. GroBschadl, Z. Hu, K. Jarvinen, H. Wang and I. Verbauwhede, "**Elliptic Curve Cryptography with Efficiently Computable Endomorphisms and Its Hardware Implementations for the Internet of Things**", IEEE Transactions on Computers, vol. 66, no. 5, pp. 773-785, 2017.
[23]Z. Liu, D. Liu and X. Zou, "**An Efficient and Flexible Hardware Implementation of the Dual-Field Elliptic Curve Cryptographic Processor**", IEEE Transactions on Industrial Electronics, vol. 64, no. 3, pp. 2353-2362, 2017.
[24]J. Mitra and T. Nayak, "**Reconfigurable very high throughput low latency VLSI (FPGA) design architecture of CRC 32**", Integration, the VLSI Journal, vol. 56, pp. 1-14, 2017.
https://doi.org/10.1016/j.vlsi.2016.09.005
[25]M. Chellam and R. Natarajan, "**AES Hardware Accelerator on FPGA with Improved Throughput and Resource Efficiency**", Arabian Journal for Science and Engineering, 2017.
[26]J. Sugier, "**Simplifying FPGA Implementations of BLAKE Hash Algorithm with Block Memory Resources**", Procedia Engineering, vol. 178, pp. 33-41, 2017.
[27]N. At, J. Beuchat, E. Okamoto, I. San and T. Yamazaki, "**A low-area unified hardware architecture for the AES and the cryptographic hash function Grøstl**", Journal of Parallel and Distributed Computing, vol. 106, pp. 106-120, 2017.
[28]S. Asif, M. Hossain, Y. Kong and W. Abdul, "**A Fully RNS based ECC Processor**", Integration, the VLSI Journal, 2017.
[29]K. Shahbazi, M. Eshghi and R. Faghih Mirzaee, "**Design and implementation of an ASIP-based cryptography processor for AES, IDEA, and MD5**", Engineering Science and Technology, an International Journal, vol. 20, no. 4, pp. 1308-1317, 2017.
[30]. J, Sasi Bhanu *et al.,"* **Enhancing Performance of IoT Networks through High Performance Computing***"* International Journal of Advanced Trends in Computer Science and Engineering, 8(3), May - June 2019, 432 – 442
https://doi.org/10.30534/ijatcse/2019/17832019
[31]J. Lee, S. Chung, H. Chang and C. Lee, "**Efficient Power-Analysis-Resistant Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture**", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 1, pp. 49-61, 2014.
[32]Z. Khan and M. Benaissa, "**High-Speed and Low-Latency ECC Processor Implementation Over GF($2^{m})$ on FPGA**", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 1, pp. 165-176, 2017.
[33]. Ziyad R. Al Ashhab *et al.*, **"Detection of HTTP Flooding DDoS Attack using Hadoop with MapReduce : A Survey"** International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.1, pp.71-77,January – February 2019
https://doi.org/10.30534/ijatcse/2019/12812019
[34].C. Lakshmi and P. J. Jayarin, "**Implementation of Krouswa Desmedt Multichiper Algorithm in ECC Processor,"** 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2019, pp. 37-40,
doi: 10.1109/ICONSTEM.2019.8918772.
[35]. Lakshmi, C. & Jayarin, P.**"Design of Flexible Multiplier Using Wallace Tree Structure for ECC Processor Over Galosis"** Field.
10.1007/978-3-030-32150-5_77.